The Intersection of Electronics, Design, and Programming

Brian Silverman

971 SPARTAN ROBOTICS

Introduction

- I’ve been involved with hardware+software+electrical on 971 for 4 years as a student and 3 as a mentor
- Various resources for individual areas, but not so much for the combination
- To me, the individual subsystems don’t have much meaning or purpose in a robot without the others
- 971 (semi-)deliberately pushes what’s possible with FRC robots
  - Tightly coupled systems like 2014 claws, 2015 elevators+arms, 2016 unfolding, 2017 turret
  - Partly happens due to getting distracted and missing the simple solution

971 SPARTAN ROBOTICS

Overview

- Some crazy things you can do
  - Going to refer back to these throughout
- Sensors
- Mechanical
- Motors

971 SPARTAN ROBOTICS

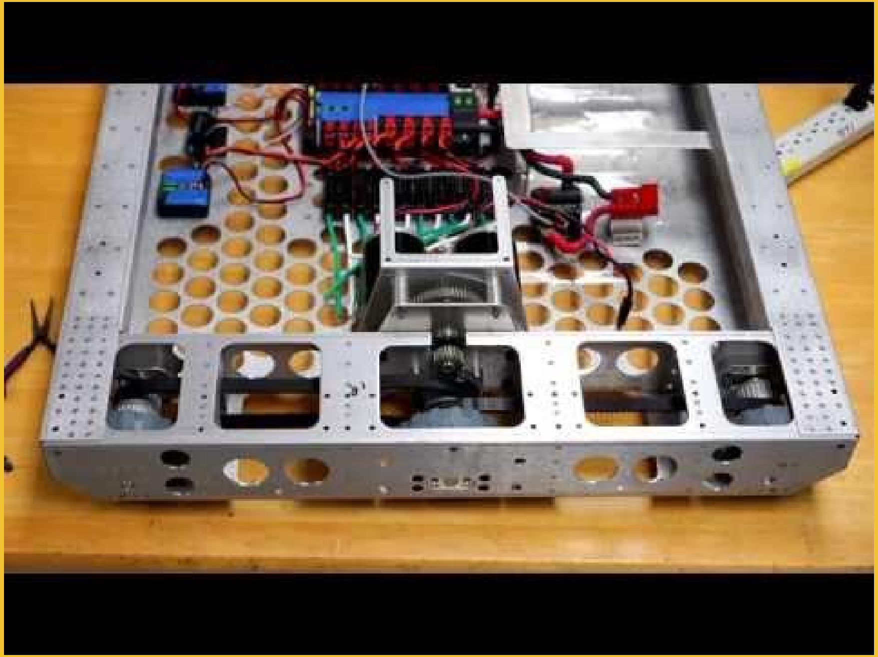
Tricky things

Holding a game piece with independent jaws

971
SPARTAN ROBOTICS

Tricky things

Moving a stack horizontally with pivoting arms



971 SPARTAN ROBOTICS

Tricky things

Unfolding a double-jointed arm in sync without colliding with itself

971 SPARTAN ROBOTICS

Tricky things

Shooting straight with a shooter mounted to the indexer

971

SPARTAN ROBOTICS

Sensors

- Most obvious place for hardware+software to interact
- Give software information about what's happening
- Going to focus on robot-internal sensors (not cameras etc)
  - Drivetrain encoders are in. Generally model information from those as saying where the robot is on a flat, obstacle-free floor rather than about what's actually around the robot
- Two relevant types: binary and rotary

971 SPARTAN ROBOTICS

Binary sensors

- On or off (aka boolean)
- Single digital input
- Easy to read
- Hall effects and switches common examples
- Switches: get triggered by acceleration
    - Some come in the KOP, many other shapes
    - 971 hasn’t used since 2012 due to bumps
- Hall effect: magnet and a sensor
    - WCP-0971 is a convenient package

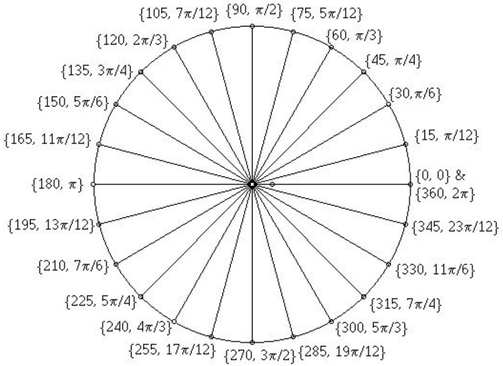
Rotary sensors

- Detect rotational position
- Absolute: know the position immediately
- Incremental: only know position relative to startup (usually need to zero, coming later)
- Limited range: 1-turn, 3-turn, etc (arms, hoods)
- Unlimited range: way too many turns to count (drivetrain, shooter wheels)
- Encoders: good precision, low noise, quick response
- Potentiometer: absolute, simple, noisy

971 SPARTAN ROBOTICS

Units

- Both one of the most important things to think about and completely irrelevant
- Easy for code to deal with pretty much anything
    - As long as you keep track of it well
- Common to work in many units for the same quantity (radians, degrees, cycles, ticks), which makes it important to specify which one
- Make sure the mechanical and software teams agree on which units they're talking about



SPARTAN ROBOTICS 971

Zeroing

- If you have an incremental encoder and want to know where the robot actually is, you have to do something
- Binary sensor at one point in the range of motion
  - Or more if you want to be fancy
- Have a potentiometer on the same mechanism
  - Sample it while everything's stationary at the start
- Encoder index pin
  - Typically more than one rotation, so still need to know which one

971 SPARTAN ROBOTICS

Zeroing in action

Robot moves both joints over hall effect sensors

Zeroing in action

Robot moves each arm over a hall effect sensor at the start of teleop

971
SPARTAN ROBOTICS

Precise software control

- Getting a mechanism to the place you want quickly and consistently takes both hardware and software
- Advanced software gets improvements, but a basic PID loop can get pretty far
- Mechanical matters more than people think
- Backlash is how much one connected thing moves without the other one
  - Motor to sensor
  - Motor to end effector
  - End effector to sensor
- Rigidity is how much things bend

971 SPARTAN ROBOTICS

Backlash

- Gears, chains, bolts, everything
- Motor ends up chattering back and forth if there's backlash there
- Backlash between the encoder and other things means you can't tell where the end effector is, so you can't make it go where you want
- Really hard to do anything about in software
- Up more reductions matters less

971 SPARTAN ROBOTICS

Rigidity

- Floppy things are hard to get where you want
- Software ends up wiggling one end back and forth, and the whole thing just flexes and doesn't go anywhere
- Think about it when designing the mechanical
- Stiffness often goes along with strength, but different materials have different properties so keep it in mind

Gear ratios

- If you're putting a lot of power through a motor all the time, it's going to get hot
- Think about holding power in addition to force
- Having extra force available lets you move it faster, and you can always limit it in software later
- More gear stages often lead to backlash though
- Think about extra friction created

Motors and controllers

- Some motors and controllers have more finesse than others
  - But also think about how much power you need
- Don’t use “brake mode”
- Motors with more cogging are harder to do fine control with
- Controllers that switch slowly are hard to work with

971 SPARTAN ROBOTICS

Takeaways

- Think about what's going to work well for software when doing mechanical and electrical
- Making everything fit well does matter
- Pick sensors deliberately
- Think about how simple you can make it first

971 SPARTAN ROBOTICS

Thank You!

971
SPARTAN ROBOTICS

Bonus slides

SPARTAN ROBOTICS 971

Encoders

- In FRC, generally means some kind of digital output
- Quadrature output (most common): incremental, easy to hook up, fast response, 2 signals
- Pulse width output: absolute, harder to read, slower/variable response time, 1 signal
- SPI/I2C (uncommon): something else with brains in the middle

971 SPARTAN ROBOTICS

Potentiometers

- Analog (continuously varying)
- Tend to be a lot noisier than encoders
- Hard to sample quickly (especially after filtering to get rid of noise)
- Fixed number of turns (1, 3, 5, 10 are common)

971
SPARTAN ROBOTICS